# WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Friday, September 17, 2004

| Hide? | Set Name | Query | Hit Count |
|---|---|---|---|
| | | *DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ* | |
| ☐ | L9 | L7 and counter | 40 |
| ☐ | L8 | L7 and hot path | 0 |
| ☐ | L7 | L5 and (branch$ or jump) | 49 |
| ☐ | L6 | L5 and L1 | 1 |
| ☐ | L5 | L3 | 132 |
| | | *DB=EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ* | |
| ☐ | L4 | L3 | 1 |
| | | *DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ* | |
| ☐ | L3 | L2 and (phase near2 (transistion or change or changing)) | 133 |
| ☐ | L2 | (edge or path) near profil$ | 11192 |
| | | *DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ* | |
| ☐ | L1 | 717/124-135,151-161.ccls. | 2661 |

END OF SEARCH HISTORY

## Web

Results 1 - 8 of 8 for **profile** **"phase change"** **"edge profiling"**. (0.53 seconds)

Tip: Try removing quotes from your search to get more results.

[DOC] Identifying Implementation-Based Testing Techniques for Classes
File Format: Microsoft Word 2000 - View as HTML
... profiling in determining **phase change** in various ... Keywords: Profiling, **edge profiling**, path profiling, performance ... FDO approaches use **profile** guided compilation ...
acis.lsfk.org/journal/format_sample.doc - Similar pages

[PDF] Representing, Detecting and Profiling Paths in Hardware
File Format: PDF/Adobe Acrobat - View as HTML
... 8: Variation in the distribution of paths caused by a **phase change** for instruction ... size, associativ- ity and the indexing scheme on the **profile** accuracy and ...
archive.csa.iisc.ernet.in/ TR/2004/2/IISc-CSA-TR-2004-2.pdf - Similar pages

[PDF] Software Profiling for Hot Path Prediction: Less is More
File Format: PDF/Adobe Acrobat
... vas@hpl.hp.com Hewlett-Packard Labs 1 Main Street Cambridge, MA 02142 ABSTRACT Recently, there has been a growing interest in exploiting **profile** information in ...
portal.acm.org/ft_gateway.cfm?id=379241&type=pdf - Similar pages

[PDF] Prefetch Injection Based on Hardware Monitoring and Object ...
File Format: PDF/Adobe Acrobat
... profiling strategies: control-flow **edge profiling** using software ... relying on dynamic **profile**-guided optimization ... optimization (ie, no **phase change**), then there ...
portal.acm.org/ft_gateway.cfm?id=996873&type=pdf - Similar pages

[PDF] Original Paper Depth **Profile** Analysis on the Nanometer Scale by a ...
File Format: PDF/Adobe Acrobat
... to the one of Auger depth **profile** analysis [4 ... The so-called crater **edge profiling** was suc- cessfully applied in ... A film of a **phase change** mate- rial, like GeSbTe ...
www.springerlink.com/index/BJ4DUJNLM5Q02ACV.pdf - Similar pages

Melting and Surface Deformation in Pulsed Laser Surface ...
... 10 μm, respectively by using knife-**edge profiling**. ... the Gaussian beam intensity **profile** along the ... and Grigoropoulos, CP, 1998, "**Phase-Change** Phenomena and ...
link.aip.org/link/?JHTRAO/122/107/1 - Similar pages

[PDF] c Copyright by Anand Shukla, 2003
File Format: PDF/Adobe Acrobat - View as HTML
... for vertex profiling, and the minimum number of edge counters for **edge profiling**. ... In a dynamic optimization system, edge **profile** or basic block **profile** can be ...
llvm.cs.uiuc.edu/pubs/2003-07-18-ShuklaMSThesis.pdf - Similar pages

[PS] cfl Copyright by Anand Shukla, 2003 LIGHTWEIGHT, CROSS-PROCEDURE ...
File Format: Adobe PostScript - View as Text
... and the minimum number of edge counters for **edge profiling**. ... or edge **profile** is also a solution to path **profile**. ... Now if there is no **phase change** in the loop ...
llvm.cs.uiuc.edu/pubs/2003-07-18-ShuklaMSThesis.ps - Similar pages

CiteSeer Find: 'phase change' AND edge profile Documents Citations

Searching for **phase change and edge profile**.
Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web) CSB DBLP
**No documents match Boolean query. Trying non-Boolean relevance query.**
500 documents found. **Only retrieving 125 documents (System busy - maximum reduced). Order: relevance to query.**

Efficient Path Profiling - Ball, Larus (1996) (Correct) (91 citations)
stalls, cache misses, or page faults. A minor **change** to the path profiling code could increment a
profiling subsumes the more common basic block and **edge** profiling, which only approximate path
www.stanford.edu/class/cs343/ps/pathprof.ps

Improved Algorithms for Dynamic Shortest Paths - Djidjev, Pantziou, Zaroliagis (1996) (Correct) (2 citations)
(but no negative cycles) that has a preprocessing **phase** during which an O(n) size data structure is
environment, where the cost of any **edge** can be **change**d or the **edge** can be deleted. In the case of
in a dynamic environment, where the cost of any **edge** can be **change**d or the **edge** can be deleted. In the
www.dcs.warwick.ac.uk/people/academic/Hristo.Djidjev/papers/dyn-spp.ps.Z

Using Constraints To Achieve Stability In Automatic Graph.. - Böhringer, Paulisch (1990) (Correct)
layout algorithm, which is divided into four **phases**: Topological Sorting: Assign nodes to levels
layout algorithm. A graph whose layout does not **change** much when it is newly layed out is called stable.
Graphs, consisting of a set of nodes and a set of **edges**, are one of the most fundamental ways of
www.ee.washington.edu/faculty/karl/Publications/PS/SIGCHI90.ps.gz

Image Parsing for Image Retrieval From Large Image Data Bases.. - Sinclair (Correct)
**edge** [13, 1] and morphological attempts [8, 12] to **phase** congruency models [5] and multi-scale approaches
as the viewpoint from which the shape is seen **changes**. They do not however map readily onto a persons
The first stage in the process is multi-scale **edge** detection. A fixed set of different sized kernels
ftp.orl.co.uk/pub/docs/ORL/tr.97.4.ps.Z

Intelligent Computing About Complex Dynamical Systems - Zhao (1994) (Correct)
systems theory and control theory, a qualitative **phase**-space representation of dynamical systems,
control objective, a control law is synthesized to **change** the natural dynamics of the system. The domain of
where nodes are intersections of flow pipes and **edges** are segments of flow pipes. The initial state and
www.cis.ohio-state.edu/insight/papers/mcs.ps

Profile Optimal 8-QAM and 32-QAM Constellations - Liu, Wesel (1998) (Correct)
labeling which maximizes the constellation **edge profile** can provide improved metrics. We
www.ee.ucla.edu/~wesel/documents/labelingweb.ps

Edge Profiling versus Path Profiling: The Showdown - Ball, Mataga, Sagiv (1998) (Correct) (18 citations)
of D[v] in Figure 3. In fact, the only **change** is in the definition of the **edge** function. In
**Edge** Profiling versus Path Profiling: The Showdown
www.bell-labs.com/user/tball/papers/popl98.ps.gz

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 47, NO. 6.. - Richard Wesel Senior (Correct)
in contrast. For pulse-amplitude modulation (PAM)**phase**-shift keying (PSK)and-QAM (square)
this is the smallest number possible.Through a **change** of basis, any such labeling has **edge** labels
constellation labeling in the context of the **edge profile**. A constellation's **edge profile** lists the
www.ee.ucla.edu/~wesel/documents/IT/Wesel01.pdf

A Numerical Algorithm Using Multizone Adaptive Grid Generation .. - Zhang Prasad (Correct)
scheme for accurate and efficient simulation of **phase change** and transport processes of industrial
for accurate and efficient simulation of **phase change** and transport processes of industrial importance.
flows in the melt. The temperature **profiles** at t =0.529h have been presented in Fig. 4(b)
thermsa.eng.sunysb.edu/~hzhang/PAPER/hui_nht96.ps

Towards Optimality in Constellation Labeling - Wesel, Komninakis, Liu (1997) (Correct)
[2] G. Ungerboeck. Channel Coding with Multilevel/**Phase** Signals. IEEE Trans. on Inform. Theory, constellations whose **edge** labels are related by a **change** of basis are distance equivalent. If two such and subsequently constellations. Section 4 uses **edge-profile** maximization to identify good
www.ee.ucla.edu/~wesel/documents/c1p5_preprint.ps

A General Empirically Based Micro-Instability.. - Vlad, Marinucci.. (1998) (Correct)
by two-scale lengths during the linear **phase** of the instabilities. A general perturbation in a DIII-D [7] has shown that the ion transport may **change** from gyro-Bohm in H-mode discharges with narrow such as the neutral dynamics at the plasma **edge**, which affects the density **profile** shape, must be
vlad.frascati.enea.it/Papers/vlad_NF_98.ps

A Quantitative Study of Differentiated Services for the.. - Sahu, Towsley, Kurose (1999) (Correct) (21 citations)
other hand, under PS, d PS h is not affected by **changes** in the non-preferred packet arrival rate. Now router mechanisms for aggregate traffic, and **edge** mechanisms for individual flows, that together can should forward packets that fall outside of the "**profile**" it has negotiated with the sender. Prior to
gaia.cs.umass.edu/pub/Sahu99_Diffserv-TR-99-09.ps.gz

Constraints on Synchronizing Oscillator Networks - David Cairns (1993) (Correct) (1 citation)
oscillator models by their underlying dynamics. **Phase** response graphs are used to determine the **phase** state of one node to the other. This causes a **change** in the period of the receiving node and therefore
www.biols.susx.ac.uk/Home/Roland_Baddeley/phase.ps.Z

Automatic Abduction of Qualitative Models - Richards, Kraan, al. (1992) (Correct) (11 citations)
generation process is broken into three major **phases**. In the first **phase** ,if we are given state 1 Variable Magnitude Direction-of-**Change** Inflow in1 steady Outflow 0 increasing Netflow
ftp.cs.utexas.edu/pub/mooney/papers/misq-aaai-92.ps.Z

Practical Estimates of the Errors Associated with the.. - Fulton, Namkung, Melvin (1992) (Correct)
equation, 1) where Df is the relative **phase change** due to displacements between two nearby equation, 1) where Df is the relative **phase change** due to displacements between two nearby points
techreports.larc.nasa.gov/pub/techreports/larc/92/conf-rpqnde-92-fulton.ps.Z

Procedure Mapping Using Static Call Graph Estimation - Hashemi, Kaeli, Calder (1997) (Correct) (1 citation)
since the unpopular procedures rarely cause a **change** in control flow. For our approach to be useful, a branches)we statically predict how often each **edge** in the call graph is traversed. These estimated of cache line conflicts. Most of these schemes use **profile** data in order to reposition the code in the
www-cse.ucsd.edu/users/calder/papers/ICCA97.ps.Z

Oscillation Phase Dynamics In The Belousov-Zhabotinsky.. - Rubin Aliev (1994) (Correct)
Oscillation **Phase** Dynamics In The Belousov-Zhabotinsky Reaction.
www.musc.edu/~alievr/papers/jpc94a_txt.ps.gz

An Overview of Document Mining Technology - Dixon (1997) (Correct)
information. Here they outline the preprocessing **phase** as a crucial one, effectively changing the nature between the IRA and car bombs? Do frequent **changes** of company management lead to better profits? data, possibly giving companies that competitive **edge** they need to survive. keywords: Document mining,
www.geocities.com/~mjdixon/mark/writings/dixm97_dm.ps

epsilon-Transformation: Exploiting Phase Transitions to.. - Zhang, Pemberton (1994) (Correct) (5 citations)
ffl-Transformation: Exploiting **Phase** Transitions to Solve Combinatorial Optimization
24, 31, 33, 34]A **phase** transition is a dramatic **change** to some problem property as some order parameter random branching factors with mean b. Nonnegative **edge** costs are bounded i.i.d. random variables. The
ftp.cs.ucla.edu/tech-report/94-reports/940003.ps.Z

Skill-Biased Technical Change and Wages: Evidence from a.. - Brian Bell (1996) (Correct) (1 citation)
Skill-Biased Technical **Change** and Wages: Evidence from a Longitudinal Data
otherwise. Notice that a standard age-earnings **profile** would 1 This contrasts with the results of
www.nuff.ox.ac.uk/economics/papers/1996/w25/computer.ps

*First 20 documents* Next 20

Terms used **profile** AND **phase change** and **hardware** and **counter**

Found **10,377** of **142,346**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

☐ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

**1** Software profiling for hot path prediction: less is more

Evelyn Duesterwald, Vasanth Bala

November 2000 **Proceedings of the ninth international conference on Architectural support for programming languages and operating systems**, Volume 34 , 28 Issue 5 , 5

Full text available: pdf(286.07 KB) Additional Information: full citation, abstract, references, citings, index terms

Recently, there has been a growing interest in exploiting profile information in adaptive systems such as just-in-time compilers, dynamic optimizers and, binary translators. In this paper, we show that sophisticated software profiling schemes that provide highly accurate information in an offline setting are ill-suited for these dynamic code generation systems. We experimentally demonstrate that hot path predictions must be made early in order to control the rising cost of missed opportunity tha ...

**2** Software profiling for hot path prediction: less is more

Evelyn Duesterwald, Vasanth Bala

November 2000 **ACM SIGPLAN Notices**, Volume 35 Issue 11

Full text available: pdf(2.43 MB) Additional Information: full citation, abstract, references, index terms

Recently, there has been a growing interest in exploiting profile information in adaptive systems such as just-in-time compilers, dynamic optimizers and, binary translators. In this paper, we show that sophisticated software profiling schemes that provide highly accurate information in an offline setting are ill-suited for these dynamic code generation systems. We experimentally demonstrate that hot path predictions must be made early in order to control the rising cost of missed opportunity tha ...

**3** Profile-based optimizations: Dynamic trace selection using performance monitoring hardware sampling

Howard Chen, Wei-Chung Hsu, Jiwei Lu, Pen-Chung Yew, Dong-Yuan Chen

March 2003 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization**

Full text available: pdf(1.88 MB) Additional Information: full citation, abstract, references, index terms

Publisher Site

Optimizing programs at run-time provides opportunities to apply aggressive optimizations to programs based on information that was not available at compile time. At run time,

programs can be adapted to better exploit architectural features, optimize the use of dynamic libraries, and simplify code based on run-time constants.Our profiling system provides a framework for collecting information required for performing run-time optimization. We sample the performance hardware registers available on ...

4 Power optimizations for cache memory: HotSpot cache: joint temporal and spatial locality exploitation for i-cache energy reduction

Chia-Lin Yang, Chien-Hao Lee

August 2004 **Proceedings of the 2004 international symposium on Low power electronics and design**

Full text available: pdf(851.58 KB) Additional Information: full citation, abstract, references, index terms

Power consumption is an important design issue of current embedded systems. It has been shown that the instruction cache accounts for a significant portion of the power dissipation of the whole chip. Several studies propose to add a cache (L0 cache) that is very small relative to the conventional L1 cache on chip for power optimization since a smaller cache has lower load capacitance. However, energy savings often come at the cost of performance degradation. In this paper, we propose a novel ins ...

**Keywords**: embedded systems, instruction cache, low power design

5 Phase tracking and prediction

Timothy Sherwood, Suleyman Sair, Brad Calder

May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture**, Volume 31 Issue 2

Full text available: pdf(674.18 KB) Additional Information: full citation, abstract, references

In a single second a modern processor can execute billions of instructions. Obtaining a bird's eye view of the behavior of a program at these speeds can be a difficult task when all that is available is cycle by cycle examination. In many programs, behavior is anything but steady state, and understanding the patterns of behavior, at run-time, can unlock a multitude of optimization opportunities.In this paper, we present a unified profiling architecture that can efficiently capture, classify, and ...

6 Compilation and run-time systems: Vacuum packing: extracting hardware-detected program phases for post-link optimization

Ronald D. Barnes, Erik M. Nystrom, Matthew C. Merten, Wen-mei W. Hwu

November 2002 **Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(1.26 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

This paper presents Vacuum Packing, a new approach to profile-based program optimization. Instead of using traditional aggregate or summarized execution profile weights, this approach uses a transparent hardware profiler to automatically detect execution phases and record branch profile information for each new phase. The code extraction algorithm then produces code packages that are specially formed for their corresponding phases. The algorithm compensates for the incomplete and often incoheren ...

7 Run-time modeling and estimation of operating system power consumption

Tao Li, Lizy Kurian John

June 2003 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems**, Volume 31 Issue 1

Full text available: pdf(233.33 KB) Additional Information: full citation, abstract, references, citings, index terms

The increasing constraints on power consumption in many computing systems point to the need for power modeling and estimation for all components of a system. The Operating System (OS) constitutes a major software component and dissipates a significant portion of total power in many modern application executions. Therefore, modeling OS power is imperative for accurate software power evaluation, as well as power management (e.g. dynamic thermal control and equal energy scheduling) in the light of ...

**Keywords**: low power, operating system, power estimation

8 Managing multi-configuration hardware via dynamic working set analysis
Ashutosh S. Dhodapkar, James E. Smith
May 2002 **ACM SIGARCH Computer Architecture News**, Volume 30 Issue 2

Full text available: pdf(1.16 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Microprocessors are designed to provide good average performance over a variety of workloads. This can lead to inefficiencies both in power and performance for individual programs and during individual phases within the same program. Microarchitectures with multi-configuration units (e.g. caches, predictors, instruction windows) are able to adapt dynamically to program behavior and enable/disable resources as needed. A key element of existing configuration algorithms is adjusting to program phas ...

9 Dynamic Adaptive compilation: Dynamic profiling and trace cache generation
Marc Berndl, Laurie Hendren
March 2003 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization**

Full text available: pdf(950.33 KB) Publisher Site Additional Information: full citation, abstract, references, index terms

Dynamic program optimization is increasingly important for achieving good runtime performance. A key issue is how to select which code to optimize. One approach is to dynamically detect traces, long sequences of instructions spanning multiple methods, which are likely to execute to completion. Traces are easy to optimize and have been shown to be a good unit for optimization.This paper reports on a new approach for dynamically detecting, creating and storing traces in a Java virtual machine. We ...

10 Positional adaptation of processors: application to energy reduction
Michael C. Huang, Jose Renau, Josep Torrellas
May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture**, Volume 31 Issue 2

Full text available: pdf(225.57 KB) Additional Information: full citation, abstract, references, citings

Although adaptive processors can exploit application variability to improve performance or save energy, effectively managing their adaptivity is challenging. To address this problem, we introduce a new approach to adaptivity: the *Positional* approach. In this approach, both the *testing* of configurations and the *application* of the chosen configurations are associated with particular code sections. This is in contrast to the currently-used *Temporal* approach to adaptation ...

11 Prefetch inection based on hardware monitoring and object metadata
Ali-Reza Adl-Tabatabai, Richard L. Hudson, Mauricio J. Serrano, Sreenivas Subramoney
June 2004 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2004 conference**

**on Programming language design and implementation**, Volume 39 Issue 6

Full text available: pdf(288.00 KB) Additional Information: full citation, abstract, references, index terms

Cache miss stalls hurt performance because of the large gap between memory and processor speeds - for example, the popular server benchmark SPEC JBB2000 spends 45% of its cycles stalled waiting for memory requests on the Itanium® 2 processor. Traversing linked data structures causes a large portion of these stalls. Prefetching for linked data structures remains a major challenge because serial data dependencies between elements in a linked data structure preclude the timely materialization ...

**Keywords**: cache misses, compiler optimization, garbage collection, prefetching, profile-guided optimization, virtual machines

12 Exploiting hardware performance counters with flow and context sensitive profiling

Glenn Ammons, Thomas Ball, James R. Larus

May 1997 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1997 conference on Programming language design and implementation**, Volume 32 Issue 5

Full text available: pdf(1.67 MB) Additional Information: full citation, abstract, references, citings, index terms

A program profile attributes run-time costs to portions of a program's execution. Most profiling systems suffer from two major deficiencies: first, they only apportion simple metrics, such as execution frequency or elapsed time to static, syntactic units, such as procedures or statements; second, they aggressively reduce the volume of information collected and reported, although aggregation can hide striking differences in program behavior.This paper addresses both concerns by exploiting the har ...

13 Runtime Power Monitoring in High-End Processors: Methodology and Empirical Data

Canturk Isci, Margaret Martonosi

December 2003 **Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture**

Full text available: pdf(921.50 KB) Publisher Site Additional Information: full citation, abstract, citings, index terms

With power dissipation becoming an increasingly vexingproblem across many classes of computer systems, measuringpower dissipation of real, running systems has becomecrucial for hardware and software system research and design.Live power measurements are imperative for studiesrequiring execution times too long for simulation, such asthermal analysis. Furthermore, as processors become morecomplex and include a host of aggressive dynamic powermanagement techniques, per-component estimates of powerd ...

14 Accurate indirect branch prediction

Karel Driesen, Urs Hölzle

April 1998 **ACM SIGARCH Computer Architecture News , Proceedings of the 25th annual international symposium on Computer architecture**, Volume 26 Issue 3

Full text available: pdf(1.49 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Indirect branch prediction is likely to become increasingly important in the future because indirect branches occur more frequently in object-oriented programs. With misprediction rates of around 25% on current processors, indirect branches can incur a significant fraction of branch misprediction overhead even though they remain less frequent than the more predictable conditional branches. We investigate a wide range of two-level predictors dedicated exclusively to indirect branches. Starting wi ...

15

The Performance of Runtime Data Cache Prefetching in a Dynamic Optimization System

Jiwei Lu, Howard Chen, Rao Fu, Wei-Chung Hsu, Bobbie Othmer, Pen-Chung Yew, Dong-Yuan Chen

December 2003 **Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture**

Full text available: pdf(253.79 KB) Publisher Site

Additional Information: full citation, abstract, citings, index terms

Traditional software controlled data cache prefetching isoften ineffective due to the lack of runtime cache miss andmiss address information. To overcome this limitation, weimplement runtime data cache prefetching in the dynamicoptimization system ADORE (ADaptive Object code RE-optimization).Its performance has been compared withstatic software prefetching on the SPEC2000 benchmarksuite. Runtime cache prefetching shows better performance.On an Itanium 2 based Linux workstation, it can increasepe ...

16 Dynamically allocating processor resources between nearby and distant ILP

Rajeev Balasubramonian, Sandhya Dwarkadas, David H. Albonesi

May 2001 **ACM SIGARCH Computer Architecture News , Proceedings of the 28th annual international symposium on Computer architecture**, Volume 29 Issue 2

Full text available: pdf(998.02 KB) Publisher Site

Additional Information: full citation, abstract, references, citings, index terms

*Modern superscalar processors use wide instruction issue widths and out-of-order execution in order to increase instruction-level parallelism (ILP). Because instructions must be committed in order so as to guarantee precise exceptions, increasing ILP implies increasing the sizes of structures such as the register file, issue queue, and reorder buffer. Simultaneously, cycle time constraints limit the sizes of these structures, resulting in conflicting design requirements.*

*In ...*

17 ProfileMe: hardware support for instruction-level profiling on out-of-order processors

Jeffrey Dean, James E. Hicks, Carl A. Waldspurger, William E. Weihl, George Chrysos

December 1997 **Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(1.60 MB) Publisher Site

Additional Information: full citation, abstract, references, citings, index terms

Profile data is valuable for identifying performance bottlenecks and guiding optimizations. Periodic sampling of a processor's performance monitoring hardware is an effective, unobtrusive way to obtain detailed profiles. Unfortunately, existing hardware simply counts events, such as cache misses and branch mispredictions, and cannot accurately attribute these events to instructions, especially on out-of-order machines. We propose an alternative approach, called ProfileMe, that samples instructio ...

18 A hardware-driven profiling scheme for identifying program hot spots to support runtime optimization

Matthew C. Merten, Andrew R. Trick, Christopher N. George, John C. Gyllenhaal, Wen-mei W. Hwu

May 1999 **ACM SIGARCH Computer Architecture News , Proceedings of the 26th annual international symposium on Computer architecture**, Volume 27 Issue 2

Full text available: pdf(349.69 KB) Publisher Site

Additional Information: full citation, abstract, references, citings, index terms

This paper presents a novel hardware-based approach for identifying, profiling, and monitoring hot spots in order to support runtime optimization of general purpose programs. The proposed approach consists of a set of tightly coupled hardware tables and control logic modules that are placed in the retirement stage of a processor pipeline removed from the critical path. The features of the proposed design include rapid detection of program hot spots after changes in execution behavior, runtime-tu ...

19 A scalable cross-platform infrastructure for application performance tuning using hardware counters

S. Browne, J. Dongarra, N. Garner, K. London, P. Mucci

November 2000 **Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM)**

Full text available: pdf(2.82 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

The purpose of the PAPI project is to specify a standard API for accessing hardware performance counters available on most modern microprocessors. These counters exist as a small set of registers that count "events", which are occurrences of specific signals and states related to the processor's function. Monitoring these events facilitates correlation between the structure of source/object code and the efficiency of the mapping of that code to the underlying architecture. This ...

20 Reconfigurable computing: architectures and applications: Using reconfigurability to achieve real-time profiling for hardware/software codesign

Lesley Shannon, Paul Chow

February 2004 **Proceeding of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays**

Full text available: pdf(228.02 KB) Additional Information: full citation, abstract, references, index terms

Embedded systems combine a processor with dedicated logic to meet design specifications at a reasonable cost. The attempt to amalgamate two distinct design environments introduces many problems, one being how to partition a single design for the two platforms to achieve the best performance with the least effort. Since the latest FPGA technology allows the integration of soft or hard CPU cores with dedicated logic on a single chip, this presents new opportunities for addressing hardware/software ...

**Keywords**: FPGA, embedded processor, hardware/software codesign, performance measurement, profiling, soft processor